A Traveling Salesman Learns Bayesian Networks

Tuhin Sahai[†] Stefan Klus[†] Michael Dellnitz[‡]

[†]United Technologies Research Center, 411 Silver Lane, East Hartford, CT 06108, USA.
[‡]University of Paderborn, Warburger Str. 100, 33098 Paderborn, Germany

Abstract

Structure learning of Bayesian networks is an important problem that arises in numerous machine learning applications. In this work, we present a novel approach for learning the structure of Bayesian networks using the solution of an appropriately constructed traveling salesman problem. In our approach, one computes an optimal ordering (partially ordered set) of random variables using methods for the traveling salesman problem. This ordering significantly reduces the search space for the subsequent greedy optimization that computes the final structure of the Bayesian network. We demonstrate our approach of learning Bayesian networks on real world census and weather datasets. In both cases, we demonstrate that the approach very accurately captures dependencies between random variables. We check the accuracy of the predictions based on independent studies in both application domains.

1 Introduction

Bayesian networks belong to the class of probabilistic graphical models and can be represented as directed acyclic graphs (DAGs) [1]. They have been used extensively in a wide variety of applications, for instance for analysis of gene expression data [2], medical diagnostics [3], machine vision [4], behavior of robots [5], and information retrieval [6] to name a few.

Bayesian networks capture the joint probability distribution of the set χ of random variables (nodes in the DAG). The edges of the DAG capture the dependence structure between variables. In particular, nodes that are not connected to one another in the DAG are conditionally independent [7].

Learning the structure of Bayesian networks is a challenging problem and has received significant attention [7, 8, 9, 10]. It is well known that given a dataset, the problem of optimally learning the associated Bayesian network structure is NP-hard [11]. Several methods to learn the structure of Bayesian networks have been proposed over the years. Arguably, the most popular and successful approaches have been built around greedy optimization schemes [9, 12]. Exact approaches for learning the structure of Bayesian networks have a scaling of $O(n2^n + n^{k+1}C(m))$, where n is the number of random variables, k is the maximum in-degree and $C(m)$ is a linear function of the data size m [13]. These approaches are based on solving a dynamic program [14]. For large Bayesian networks the above scaling for exact algorithms is prohibitive [14].

In this work, we present a heuristic approach for learning the structure of Bayesian networks from data. The approach is based on computing an ordering of the random variables using the traveling salesman problem (TSP). Though using the ordering to learn Bayesian networks is not new [15], using the TSP for this task is novel. This approach provides us with the opportunity to leverage efficient implementations of TSP algorithms such as the Lin-Kernighan heuristic[1] [17] and cutting plane methods[2] [19] for fast structure learning of Bayesian networks.

The remainder of the paper is organized as follows. In section 2, we describe the approach for learning Bayesian networks using a history dependent TSP formulation. In section 3, we develop techniques for solving the history dependent TSP. We then present results on the Adult and El Niño datasets in section 4. We finally draw conclusions and discuss future work in section 5.

2 Structure Learning of Bayesian Networks Using the Traveling Salesman Problem

Although we use the K2 metric [10] to construct the Bayesian network, the only assumption our approach makes is that the scoring metric is decomposable [14],

$$\text{GRAPHSCORE} = \sum_{x \in V} \text{NODESCORE}(x|parents(x)). \tag{1}$$

[1] LKH software [16] is a popular implementation of this approach

[2] Concorde TSP solver [18] is an efficient implementation of a cutting plane approach coupled with other heuristics

Thus, one can replace the K2 metric with any of the competing scoring functions such as BIC [20], BDeu [21], BDe [22], and minimum description length [23].

A link between the optimal ordering and the TSP can be established on the basis of the decomposable metric. To find the best possible ordering $\mathbb{O}$ we start from an empty set ϕ. We define the cost of going from ϕ to single random variables to be 0. Similarly, the cost of going from any permutation of all random variables to ϕ is also defined to be 0. For any partial ordering of random variables $\tilde{\mathbb{O}}$ (one that does not include all random variables) we know that,

$$\mathrm{V}(\tilde{\mathbb{O}}) = \mathrm{V}(\tilde{\mathbb{O}} \setminus X) + \mathrm{Cost}(X, \tilde{\mathbb{O}} \setminus X), \tag{2}$$

where X is a random variable, V is the value function, $\tilde{\mathbb{O}} \setminus X$ is the set $\tilde{\mathbb{O}}$ without X, and $\mathrm{Cost}(X, \tilde{\mathbb{O}} \setminus X)$ is the cost of adding X to $\tilde{\mathbb{O}} \setminus X$.

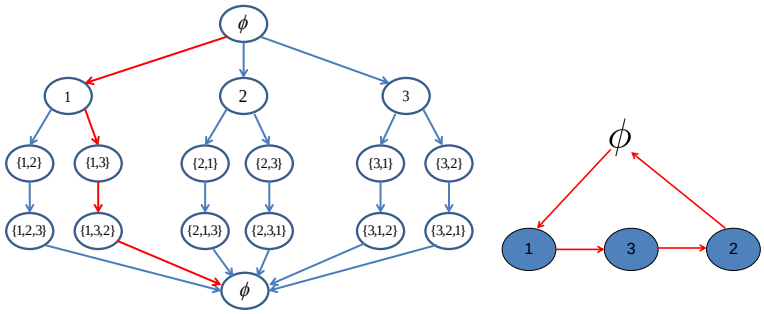
The above dynamic program in Eqn. 2 will require $O(n^2 2^n)$ operations [14]. Instead of solving the above equation using dynamic programming, we reformulate the problem as a history dependent TSP. This is easy to see from Eqn. 2, by considering the random variables as cities of the tour and the optimal ordering of random variables as a tour that minimizes the overall cost (see Eqn. 3 and Fig. 1).

$$\mathrm{V}(\mathbb{O}) = \min \sum_{i=1}^{N} \left[\mathrm{V}(\tilde{\mathbb{O}}((i+1))) - \mathrm{V}(\tilde{\mathbb{O}}(i)) \right], \tag{3}$$

The history dependence arises due to the first term in the right hand side of Eqn. 2. The advantage of treating this minimization as a TSP, however, is the ability to leverage pre-existing TSP algorithms such as LKH [16], as discussed in the next section. Note that our approach provides Bayesian networks in which the directionality of arrows (causality) may be reversed. This may be attributed to the fact that, given the data, these networks are equally likely [24].

3 Solving the History Dependent Traveling Salesman Problem

The traveling salesman problem (TSP) is a classic problem that has received attention from the applied mathematics and computer science communities



(a) (b)

Figure 1: a) Structure Learning of Bayesian networks as a dynamic program [14]. The permutation tree provides the order in which nodes should be added to the list. b) The equivalent solution of the history dependent TSP for the computation of the optimal ordering.

for decades. In the traditional formulation, one is given a list of city positions and tasked with finding a Hamiltonian cycle (a cycle that visits every city only once and returns to the starting city) with lowest cost [25]. Enumerating all possible tours becomes infeasible for problems with more than 10 cities. In particular, the TSP is a well studied NP-hard problem [26]. Over several decades, many algorithms for computing the solution of the TSP have been developed; for an overview we refer the reader to [19, 26].

To solve the history dependent TSP, we pick Helsgaun's popular version of the Lin-Kernighan Heuristic (LKH) [16], which naturally extends to our case. LKH is a randomized approach that picks edges in the tour for removal and adds ones that are "more likely" to be in the optimal tour. If the replacement of edges reduces the cost, the change to the tour is accepted. The likelihood of any edge being in the optimal tour is computed using the α-nearness that is based on minimum 1-trees in the underlying city graph [17]. The LKH is the most successful approach for computing the optimal tour of TSPs with asymmetric cost [16].

In general, one replaces k edges in a simple iteration (known as k-opt steps). Examples of the 2-opt and 3-opt steps are shown in Fig. 2. Note that using higher values of k, in general, will give tours will lower cost. However, as k increases, closing the tour becomes increasingly challenging [16].

The above approach extends naturally to the history dependent TSP.

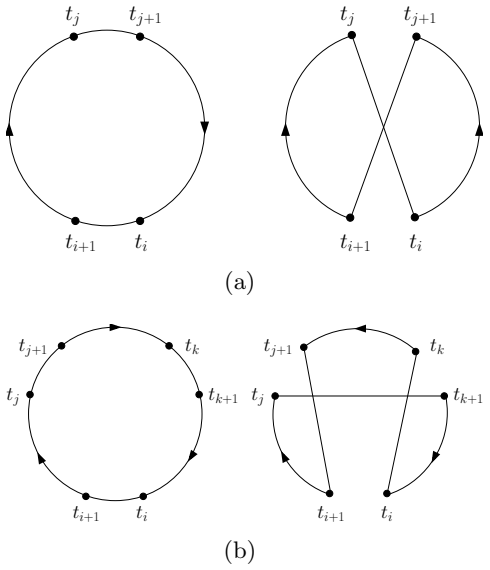


Figure 2: a) 2-opt moves for the TSP. b) 3-opt moves for the TSP.

In our problem, edges again are deleted and added randomly. Unlike the standard TSP, the acceptance or rejection of the edge replacement is now dependent on the direction as well as on the existing tour. For structure learning of Bayesian networks, we compared the 2-opt and 3-opt iterations with Helsgaun's implementation of LKH [16]. We find that despite ignoring history, the standard LKH software performs significantly better than our 2-opt and 3-opt implementations with history. This is, perhaps, due to the fact that LKH uses sequential 5-opt steps as a basic move [16] which is found to provide significantly better results. If Helsgaun's LKH software were to be integrated with history dependent costs, it would be expected to provide more accurate results. This is currently part of our future efforts at improving this approach. Thus, all results presented here were computed simply by using the LKH software.

4 Results

We now test our approach of computing the structure of Bayesian networks using the history dependent TSP on the Adult and El Niño datasets, available publicly from the UCI Machine Learning Repository [27].

4.1 Adult Dataset

The Adult dataset was extracted from census data in 1994 by Ronny Kohavi and Barry Becker [27]. The dataset consists of data for 48842 individuals and includes several attributes including occupation, salary, number of hours worked per week, race, native country, education, marital status etc. For a complete list of attributes see [27]. Unfortunately, the dataset has missing values i.e. entries for certain individuals are not available. We discard these data points to finally obtain a dataset with 30162 entries. We break this dataset into training (29162 entries) and testing (1000 entries) parts. Some of the attributes such as salary and capital gain are continuous; we discretize these attributes (for the number of possible states see table 1). We then construct a Bayesian network using our TSP and greedy hill climbing approach (shown in Fig. 3).

Work Class	7	Education	16
Marital Status	7	Occupation	14
Race	5	Capital Gain	3
Capital Loss	3	Hours/Week	3
Native Country	41	Salary	2

Table 1: Number of states for each random variable in the Adult dataset. Continuous variables have been discretized.

The Bayesian network that is learnt using the TSP and hill climbing in Fig. 3 automatically captures dependencies that are now known as a result of several independent studies. For example, the Bayesian network captures the dependency between the occupation and the number of hours worked per week [28]. Similarly, the Bayesian network in Fig. 3 predicts dependencies between education and salary [29], marital status and salary [30], occupation and race [31], and marital status and number of hours of worked per week [32]. The dependencies between race and native country, occupation and class of work, and salary and hours of work are obvious by definition and simple arithmetic respectively. Thus, we believe that the approach accurately captures the dependencies between random variables from raw data without any prior knowledge of their ordering. If one inputs an incorrect ordering of random variables, the quality of predicted dependencies degrades significantly. The comparison of resulting Bayesian networks can be performed using the log likelihood ratio [33].

To quantitatively test the prediction of the resulting Bayesian network we check the prediction of $P(\text{Salary}|\text{Education}, \text{Marital Status})$. In particular,

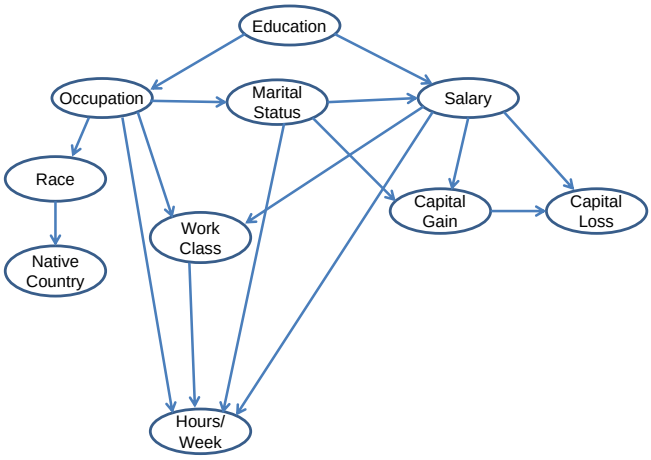


Figure 3: Structure of the Adult dataset Bayesian network learnt using the history dependent TSP and greedy hill climbing.

we check the accuracy of the dependence structure on predicting whether individuals in the testing dataset earn more than or less than $\$50,000$ per year (thus salary takes binary states: 0 or 1). We compute the mean square error using the following expression,

$$\text{MSE} = \frac{1}{N_t}(\mathbb{E}(Y) - Y)^2, \tag{4}$$

where N_t is the number of data points in the testing set, Y is the output state (salary in this example), and $\mathbb{E}(Y)$ is the expected value of Y predicted by the Bayesian network. The MSE for the adult dataset is 0.13. If one were to threshold probabilities at 0.5, i.e. if the $P(\text{Salary}|\text{Education}, \text{Marital Status}) > \$50,000$ is greater than 0.5, we assume Salary $= \$50,000$. In this case we find that our approach correctly predicts the salary 78% of the time.

4.2 El Niño Dataset

We now apply our algorithm to the El Niño dataset from the UCI Machine Learning Repository [27]. The data set consists of oceanographic and meteorological readings taken by buoys in the Pacific Ocean. This large dataset consists of variables such as latitude, longitude, date, zonal winds and humidity (for a complete list of variables see [27]). In this example, we try to

answer the question of dependence of variables that was posed in [27]: How do the variables relate to each other?

Just like in the Adult dataset example, we remove data points with missing values and partition the states into discrete values (see table 2). After data clean up, the dataset has 93935 data points that are used to learn the Bayesian network. We again partition the entire dataset into training (with 92935 entries) and testing (with 1000 entries) parts. The resulting Bayesian network is highly interconnected as seen in Fig. 4. In particular, we find dependencies between air temperature and humidity, air temperature and sea surface temperature, and zonal winds and air temperature. As one would expect, we find dependencies between seasons and sea surface temperature, humidity and sea surface temperature. Note that though the predicted dependencies between seasons and longitude/latitude seem peculiar, it is to be expected since the buoys were not anchored at fixed locations and were free to drift around [27]. Previous analysis of this dataset considered only correlations and failed to pick up links between zonal/meridional winds and meteorological quantities. We, however, do find dependencies between the winds and meteorological quantities, suggesting a nonlinear relationship between random variables.

Season	4	Latitude	2
Longitude	2	Zonal Wind	2
Meridonal Wind	2	Humidity	2
Air Temperature	2	Sea Surface Temperature	2

Table 2: Number of states for each random variable in the El Niño dataset. Continuous variables have been discretized.

To quantitatively test the predictions of the Bayesian network in Fig. 4, we concentrate on predicting zonal wind speeds using seasons and longitude. Using Eqn. 4, we find that predicted MSE is 0.09. If we again threshold the predicted values of zonal wind at $P > 0.5$, we find that the zonal wind is predicted with 89% accuracy.

5 Conclusions and Future Work

In this work, we have presented a new approach for learning the structure and parameters of a Bayesian network. The method computes an ordering of the random variables based on a history dependent TSP on the random variables. This ordering, typically supplied by domain knowledge experts,

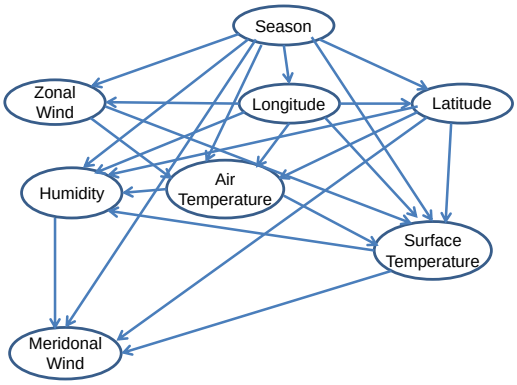


Figure 4: Structure of the El Niño dataset Bayesian network learnt using the history dependent TSP and greedy hill climbing.

significantly reduces the search space for hill-climbing methods. This makes the underlying optimization techniques effective at finding Bayesian network structures that maximize likelihood.

For computing the solution of the TSP, we use the Lin-Kernighan heuristic [17, 16] with history dependent cost. The LKH approach is shown to extend naturally to this case. We used the TSP with greedy hill climbing to compute Bayesian networks to analyze the publicly available Adult and El Niño datasets [27]. We find that the approach successfully computes Bayesian networks that accurately capture the underlying system interdependencies. We check the results against common knowledge as well as domain specific studies.

Future work includes the development of novel and fast heuristics for the history dependent TSP. There is a significant lack of methods to deal with this class of problems. Additionally, to provide scalability, the authors are investigating the utility of decentralized clustering methods [34] to learn Bayesian networks in distributed settings.

6 Acknowledgements

The authors thank Madhu Shashanka of UTRC for suggestions and discussions related to this work.

References

[1] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, first edition, 2009.

[2] N. Friedman, M. Linial, I. Nachman, and D. Peér. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3-4):601–620, 2000.

[3] D. Heckerman, E. Horvitz, and B. Nathwani. Toward normative expert systems: Part I. The pathfinder project. *Methods of Information in Medicine*, 31:90–105, 1992.

[4] T. S. Levitt, J. M. Agosta, and T. O. Binford. Model-based influence diagrams for machine vision. In *Proceedings of the Fifth Annual Conference on Uncertainty in Artificial Intelligence*, UAI '89, pages 371–388, Amsterdam, The Netherlands, 1990. North-Holland Publishing Co.

[5] E. Lazkano, B. Sierra, A. Astigarraga, and J. M. Martínez-Otzeta. On the use of Bayesian networks to develop behaviours for mobile robots. *Robotics and Autonomous Systems*, 55(3):253–265, March 2007.

[6] R. Fung and B. Del Favero. Applying Bayesian networks to information retrieval. *Communications of the ACM*, 38(3):42–ff., March 1995.

[7] D. Heckerman. A tutorial on learning with Bayesian networks. In D. Holmes and L. Jain, editors, *Innovations in Bayesian Networks*, volume 156 of *Studies in Computational Intelligence*, pages 33–82. Springer Berlin / Heidelberg, 2008.

[8] R. E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, first edition, 2003.

[9] N. Friedman, I. Nachman, and D. Peér. Learning Bayesian network structure from massive datasets: The "sparse candidate" algorithm. pages 206–215, 1999.

[10] G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, October 1992.

[11] D. M. Chickering. Learning Bayesian networks is NP-complete. In *Learning from Data: Artificial Intelligence and Statistics V*, pages 121–130. Springer-Verlag, 1996.

[12] D. M. Chickering. Learning equivalence classes of Bayesian-network structures. *Journal of Machine Learning Research*, 2:445–498, March 2002.

[13] M. Koivisto and K. Sood. Exact Bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research*, 5:549–573, 2004.

[14] A. P. Singh and A. W. Moore. Finding optimal Bayesian networks by dynamic programming. *CMU Technical Report*, 2005.

[15] M. Teyssier and D. Koller. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. *arXiv preprint arXiv:1207.1429*, 2012.

[16] K. Helsgaun. An effective implementation of the Lin-Kernighan traveling salesman heuristic. DATALOGISKE SKRIFTER (writings on computer science), no. 81, Roskilde University, 1998.

[17] S. Lin and B. W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2):498–516, 1973.

[18] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. Concorde TSP Solver. `www.keck.caam.rice.edu/concorde.html`, 2006.

[19] G. Reinelt. *The traveling salesman: Computational solutions for TSP applications*. Springer-Verlag, Berlin, Heidelberg, 1994.

[20] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

[21] W. Buntine. Theory refinement on Bayesian networks. In *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence*, UAI'91, pages 52–60, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.

[22] D. Heckerman and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. In *Machine Learning*, pages 20–197, 1995.

[23] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465 – 471, 1978.

[24] P. Spirtes, C. Glymour, and R. Scheines. *Causation, prediction, and search*, volume 81. MIT press, 2001.

[25] R. M. Karp. Reducibility among Combinatorial Problems. *Complexity of Computer Computations*, 1972.

[26] W.J. Cook. *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation*. Princeton University Press, first edition, 2012.

[27] A. Frank and A. Asuncion. UCI machine learning repository. `http://archive.ics.uci.edu/ml`, 2010.

[28] N.W. Burton, G. Turrell, et al. Occupation, hours worked, and leisure-time physical activity. *Preventive Medicine*, 31(6):673, 2000.

[29] M. Blaug. The correlation between education and earnings: What does it signify? *Higher Education*, 1(1):53–76, 1972.

[30] A. Ahituv and R. I. Lerman. How do marital status, work effort, and wage rates interact? *Demography*, 44(3):623–647, 2007.

[31] S.M. Carlson. Trends in race/sex occupational inequality: Conceptual and measurement issues. *Social Problems*, pages 268–290, 1992.

[32] H.J. Choi, S. Lundberg, and J. Joesch. Work and family: Marriage, children, child gender and the work hours and earnings of West German men. *IZA Discussion Paper No. 1761*, 2005.

[33] J. Neyman and E. S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933.

[34] Tuhin Sahai, Alberto Speranzon, and Andrzej Banaszuk. Hearing the clusters of a graph: A distributed algorithm. *Automatica*, 48(1):15 – 24, 2012.